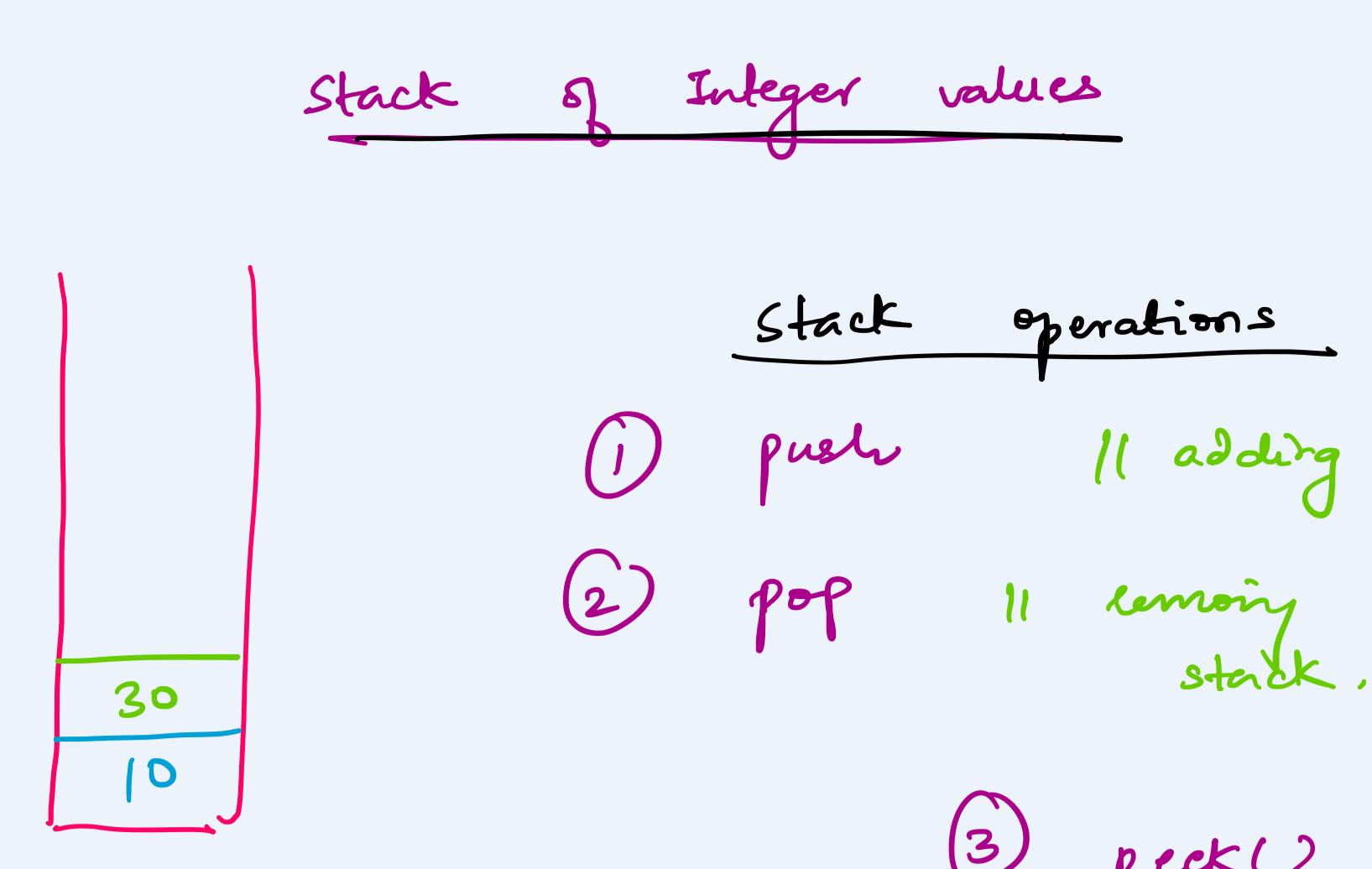
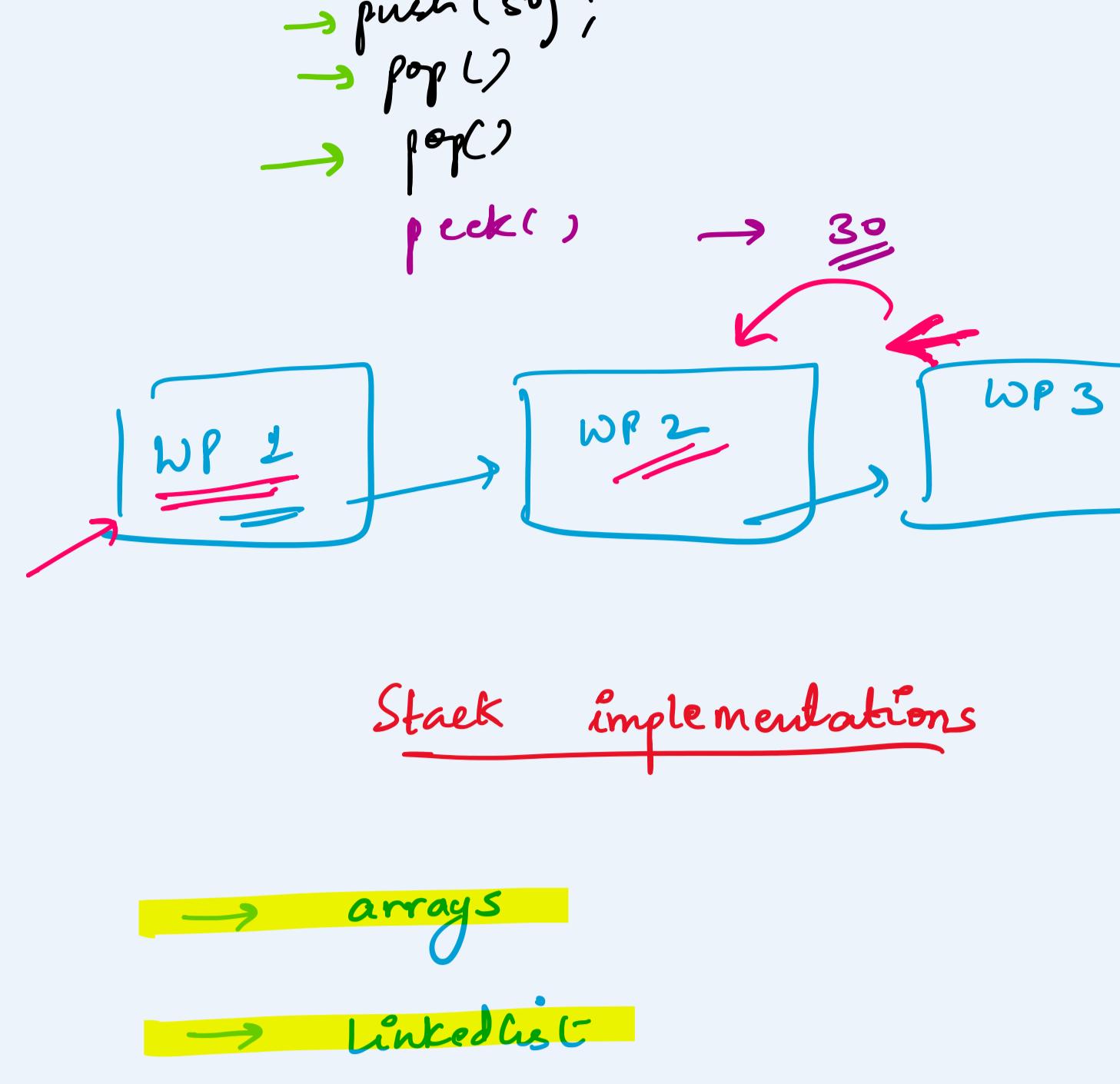


StacksStack of Integer values

push(10);
push(20);
pop();
push(30);
push(10);
push(50);
pop();
pop();
pop();
push(30);
WP 1 → WP 2 → WP 3

Stack implementations

→ arrays

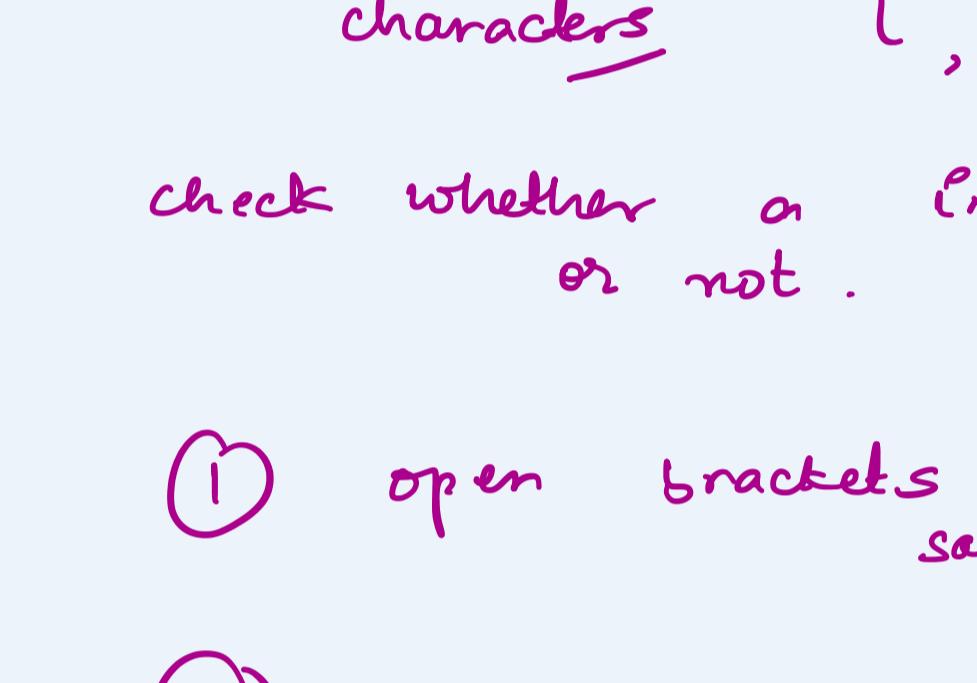
→ linkedlist

→ queues

① Arrays

[5]

LIFO



→ O(1)

→ O(1)

Stack overflow

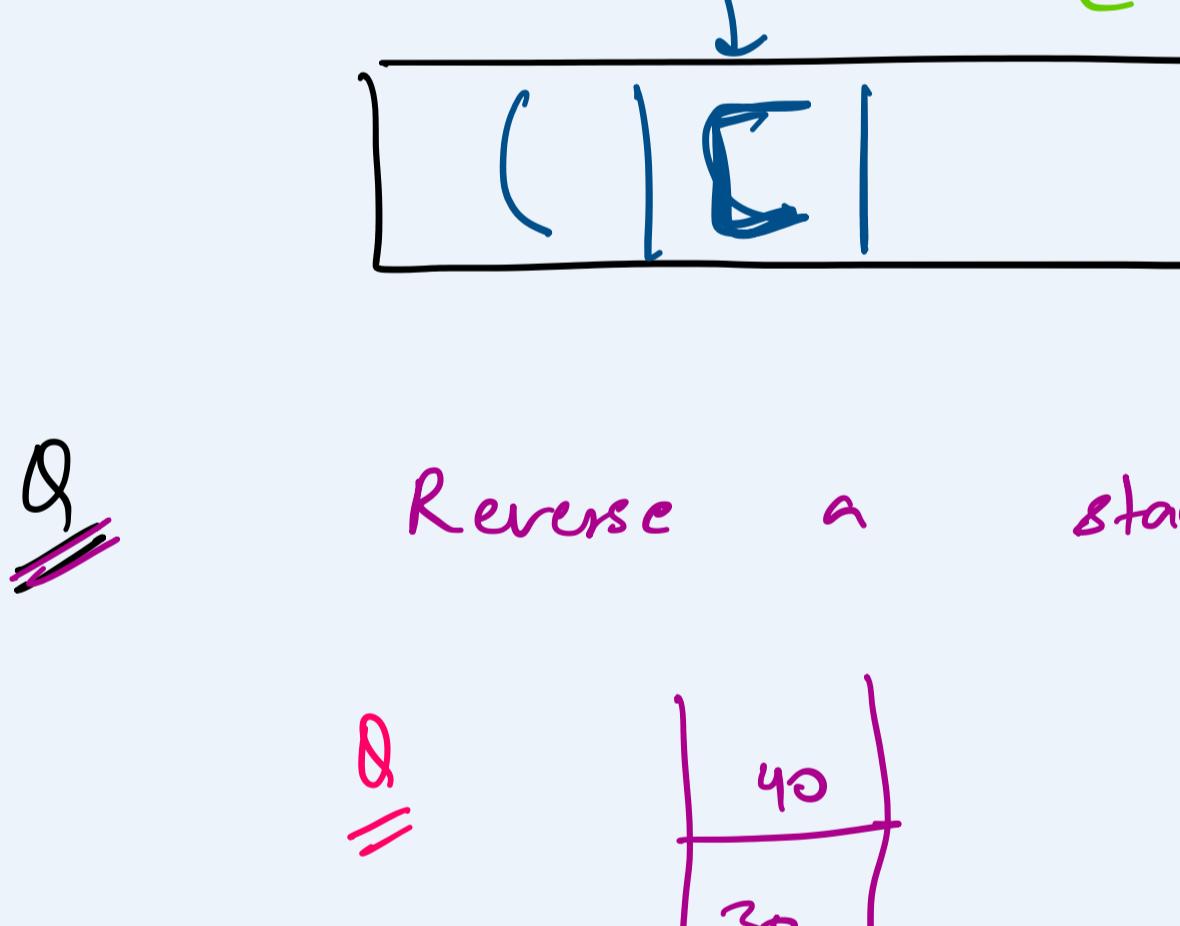
s.pop() × 6

Pros → ① → lesser space
② ;

Cons

② linkedlist

O(1) → [s.push()] → [add at head]
O(1) → [s.pop()] → [remove at head]

→ s.push(10)
→ s.push(20)
→ s.push(30)
→ s.pop();
→ s.push(40);
→ s.push(50);
→ s.push(60);
→ s.pop();
→ s.pop();

Pros : dynamize size
→ O(1), O(1)

Ques : Given a string, containing just of characters {, }, [,] , (,)
check whether a input string is valid or not.

① open brackets must be closed by the same type of brackets

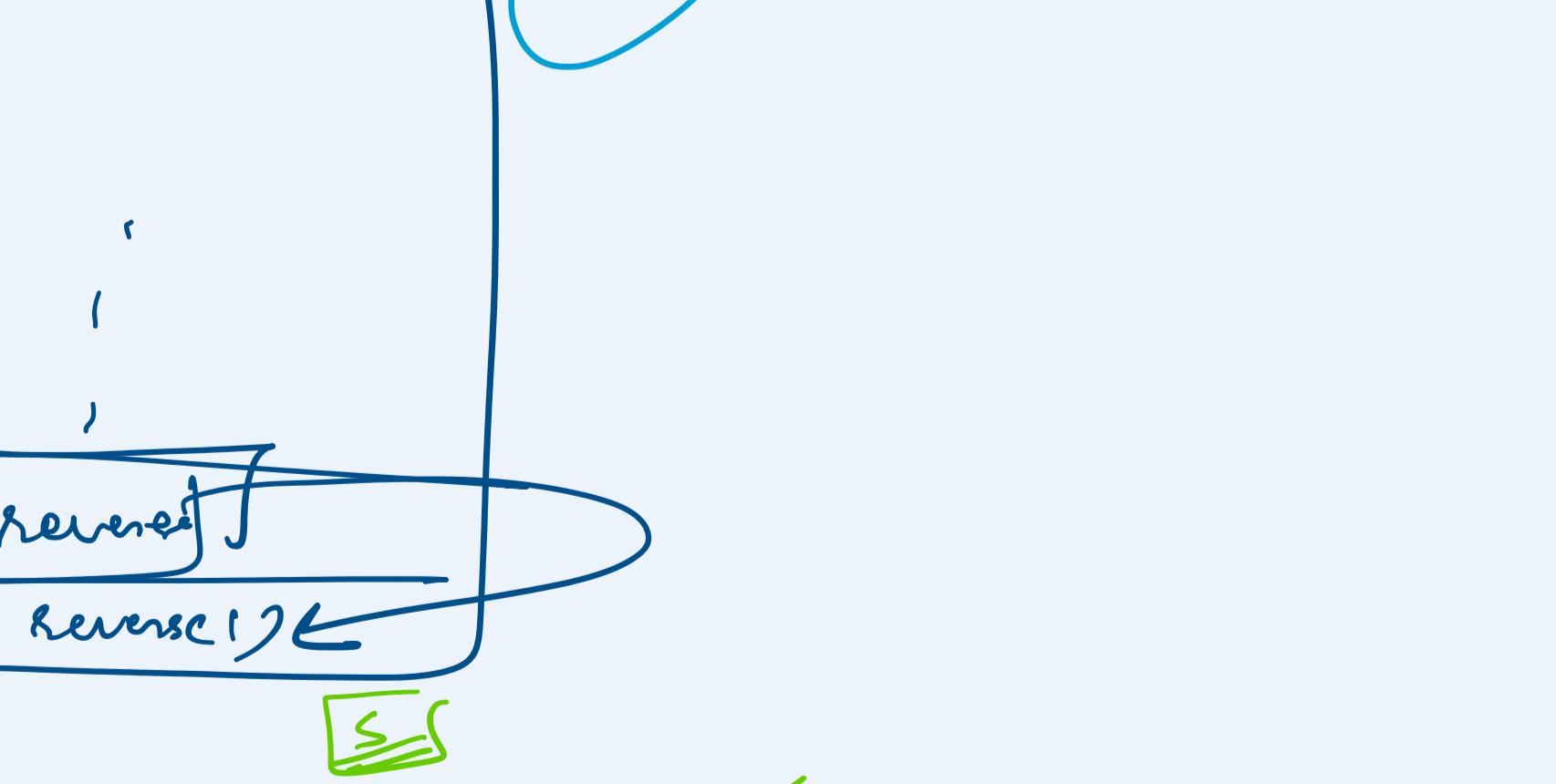
② open brackets must be closed in correct order

Eg : s = "([])" true

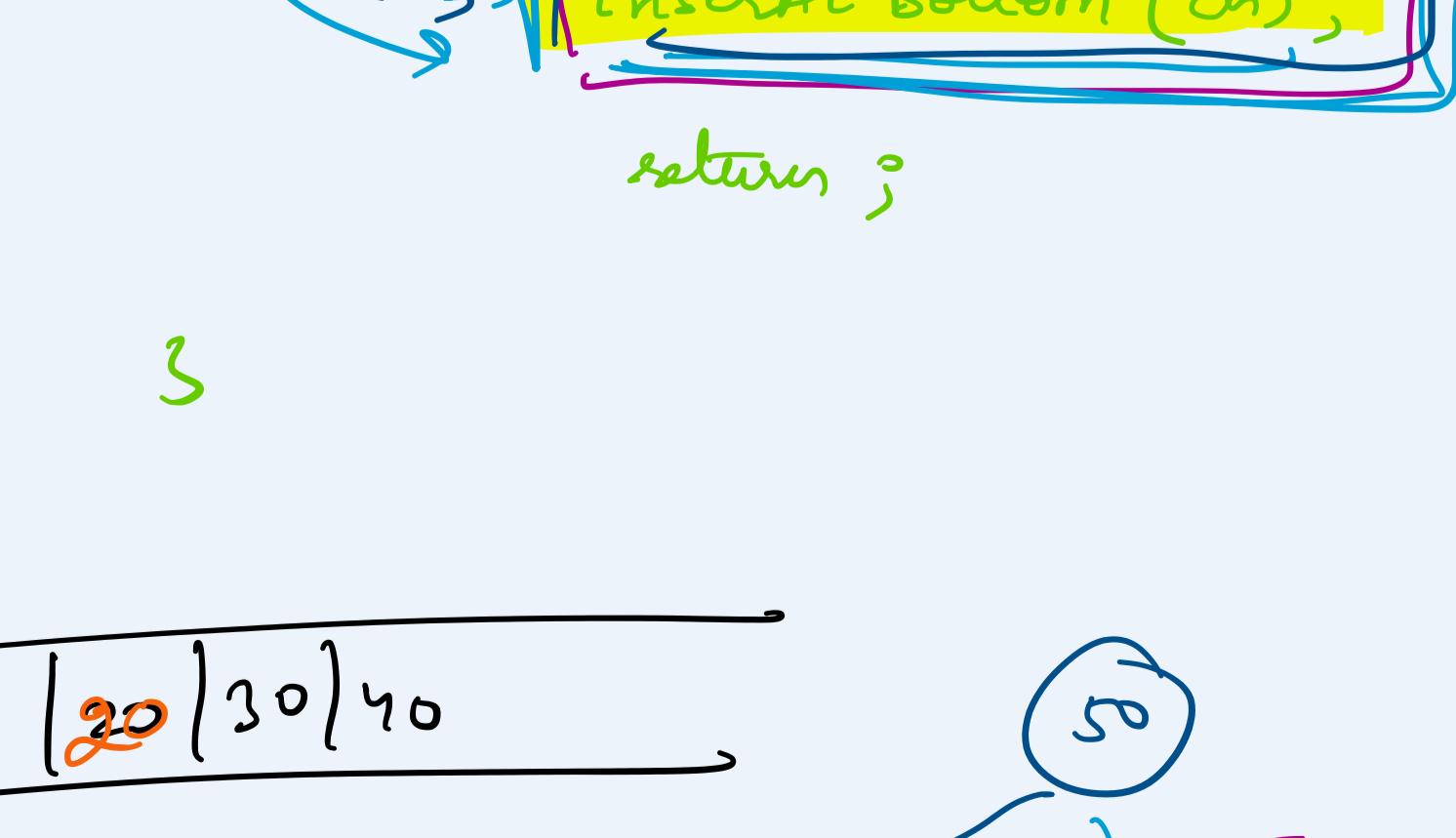
s = "({[()]}{})" true

s = "[{ }]" false

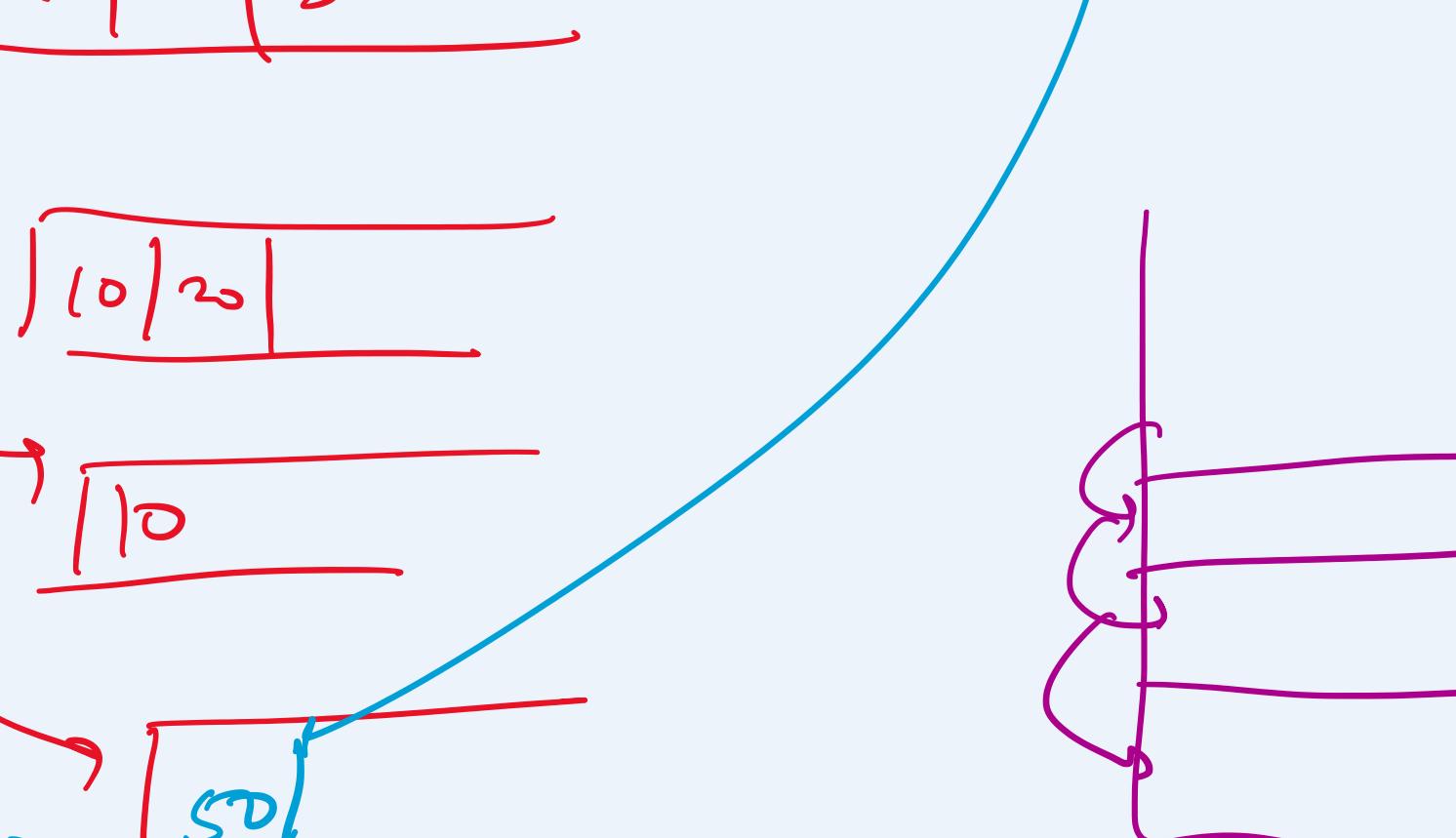
s = "[]" false



reverse(s)



int val = s.pop(); helper.push(val);



char ch = s.pop(); reverse();

insertAtBottom(ch);

return s;

